TRƯỜNG ĐẠI HỌC THĂNG LONG

**KHOA TOÁN TIN**

-----🙞🙜🕮🙞🙜-----



**BÀI TẬP LỚN**

**XÂY DỰNG HỆ THỐNG QUẢN LÝ THỜI KHÓA BIỂU**

|  |  |
| --- | --- |
| **GIÁO VIÊN HƯỚNG DẪN** | **SINH VIÊN THỰC HIỆN** |
| - ThS. Mai Thúy Nga | - A29610 - Tạ Minh Nhật  - A29208 - Nguyễn Tiến Dũng  - A28563 - Đào Tuấn Trung  - A30015 - Nguyễn Hồng Quân  - A28832 - Nguyễn Quốc Việt  **Ngành: Khoa học máy tính** |

|  |
| --- |
| **HÀ NỘI – 2019** |

TRƯỜNG ĐẠI HỌC THĂNG LONG

**KHOA TOÁN TIN**

-----🙞🙜🕮🙞🙜-----



**BÀI TẬP LỚN**

**XÂY DỰNG HỆ THỐNG QUẢN LÝ THỜI KHÓA BIỂU**

|  |  |
| --- | --- |
| **GIÁO VIÊN HƯỚNG DẪN** | **SINH VIÊN THỰC HIỆN** |
| - ThS. Mai Thúy Nga | - A29610 - Tạ Minh Nhật  - A29208 - Nguyễn Tiến Dũng  - A28563 - Đào Tuấn Trung  - A30015 - Nguyễn Hồng Quân  - A28832 - Nguyễn Quốc Việt  **Ngành: Khoa học máy tính** |

|  |
| --- |
| **HÀ NỘI – 2019** |

**LỜI CẢM ƠN**

Em xin chân thành cảm ơn!

Hà Nội, tháng năm 20

*Sinh viên thực hiện*

Họ và tên sinh viên

**MỤC LỤC**

[CHƯƠNG 1: TỔNG QUAN HỆ THỐNG 3](#_Toc20407298)

[1.1. Giới thiệu bài toán 3](#_Toc20407299)

[1.2. Yêu cầu bài toán 3](#_Toc20407300)

[1.3. Yêu cầu nghiệp vụ 4](#_Toc20407301)

[1.4. Sơ đồ tổng quan chức năng của hệ thống 5](#_Toc20407302)

[1.5. Mô tả chức năng của hệ thống 8](#_Toc20407303)

[1.6. Ánh xạ yêu cầu nghiệp vụ và chức năng hệ thống 10](#_Toc20407304)

[1.7. Lịch trình thực hiện 11](#_Toc20407305)

[CHƯƠNG 2: KIẾN TRÚC TỔNG THỂ, CÁC CÔNG NGHỆ SỬ DỤNG 12](#_Toc20407306)

[2.1. Kiến trúc tổng thể 12](#_Toc20407308)

[2.2. Công nghệ sử dụng 13](#_Toc20407309)

[CHƯƠNG 3: Thiết kế API, Database 23](#_Toc20407310)

[3.1. Thiết kế kiến trúc hệ thống API 23](#_Toc20407312)

[3.2. Danh sách API sử dụng trong hệ thống: 25](#_Toc20407313)

**DANH MỤC HÌNH ẢNH, SƠ ĐỒ**

**DANH MỤC TỪ VIẾT TẮT**

|  |  |  |
| --- | --- | --- |
| **STT** | **Từ viết tắt** | **Ý nghĩa** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# CHƯƠNG 1: TỔNG QUAN HỆ THỐNG

## Giới thiệu bài toán

**PHẦN MỀM XẾP THỜI KHÓA BIỂU**

Để có thể đăng ký học, các bộ phận của nhà trường phải phối hợp với nhau để tạo ra một thời khóa biểu toàn trường trước mỗi kì của năm học. Trong thực tế, với việc xây dựng được một thời khóa biểu hoàn chỉnh và hợp lý làm mất rất nhiều thời gian và công sức của những người chịu trách nhiệm thực hiện. Chính vì vậy bài toán sắp xếp thời khóa biểu luôn là một bài toán khó, mang tính khoa học đồng thời tính thực tiễn cũng rất cao.

Với vai trò là một người phát triển phần mềm, nhóm xây dựng một phần mềm để tin học hóa việc làm thời khóa biểu. Phần mềm đáp ứng được việc quản lý thời khóa biểu cho tất cả các ngành học trong trường theo từng kì và nhóm học (ví dụ tại đại học Thăng Long được chia làm 3 kì học, mỗi kì học lại chia thành 3 nhóm học).

Phần mềm xếp thời khóa biểu sẽ đảm bảo một thời khóa biểu “đúng” với sự sắp xếp tối ưu nhất có thể để các môn học không bị trùng giờ. Dữ liệu được lưu trữ dùng cho nhiều việc khác trong nhà trường. Phần mềm sẽ giúp ích được ít nhiều phòng đào tạo trong việc quản lý thời khóa biểu toàn trường một cách dễ dàng hơn.

## Yêu cầu bài toán

Phần mềm xếp thời khóa biểu của nhóm có thể chạy trên được trên cả 2 nền tảng là Web và Mobile (hệ điều hành IOS). Một số chức năng chính của phần mềm xếp thời khóa biểu bao gồm: chuẩn bị dữ liệu cho thời khóa biểu, tạo thời khóa biểu, sử dụng thời khóa biểu

***1.2.1 Chuẩn bị dữ liệu cho thời khóa biểu:***

Một thời khóa biểu hoàn chỉnh cần rất nhiều yếu tố liên quan, kết hợp với nhau tạo thành. Chính vì vậy mà việc chuẩn bị dữ liệu cho thời khóa biểu là hết sức quan trọng. Những dữ liệu cần chuẩn bị cho công việc tạo thời khóa biểu toàn trường như: dữ liệu về học phần, dữ liệu về phòng học, dữ liệu về giáo viên, …Thông tin về lớp bao gồm: học phần, phòng học, ca học, giáo viên dạy, số lượng sinh viên tối đa trong một lớp.

***1.2.2 Tạo thời khóa biểu:***

Sau khi đã có dữ liệu các phần đầy đủ thì công việc tiếp theo là kết hợp được các dữ liệu đó lại thành một thời khóa biểu đầy đủ và chính xác. Người mở lớp của từng bộ môn sẽ tạo thời khóa biểu cho các học phần trong phạm vi phụ trách của bộ môn. Phòng đào tạo sẽ quản lý về các dữ liệu và tổng hợp lại những yêu cầu tạo thời khóa biểu của từng bộ môn để tạo thành một thời khóa biểu toàn trường.

***1.2.3 Sử dụng thời khóa biểu:***

Giáo viên có thể xem được thời khóa biểu toàn trường hoặc là thời khóa biểu của cá nhân, của bộ môn tùy theo vào mục đích của mình. Hệ thống sẽ hiển thị view thời khóa biểu tùy theo vào đối tượng và mục đích sử dụng.

## Yêu cầu nghiệp vụ

* **BR1:** Đăng nhập
* **BR2:** Người dùng

**+ BR2.1:** Quản lý thông tin người dùng

**+ BR2.2:** Phân quyền người dùng

* **BR3:** Thời khóa biểu

**+ BR3.1:** Xếp thời khóa biểu

**+ BR3.2:** Quản lý học phần

**+ BR3.3:** Quản lý kỳ học

**+ BR3.4:** Quản lý nhóm học

**+ BR3.5:** Quản lý phòng học

**+ BR3.6:** Quản lý giáo viên

**+ BR3.7:** Quản lý thời khóa biểu

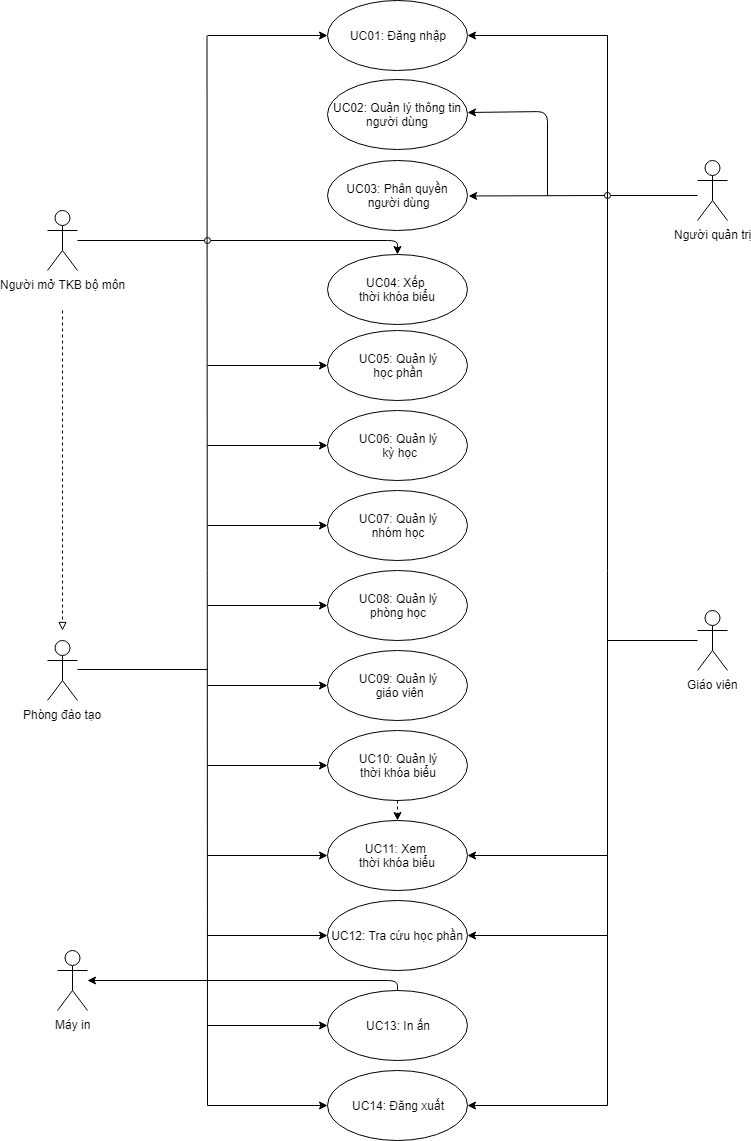
**+ BR3.8:** Xem thời khóa biểu

**+ BR3.9:** Tra cứu học phần thời khóa biểu

* **BR4:** In ấn
* **BR5:** Đăng xuất

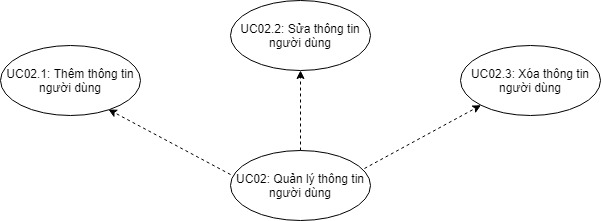
## Sơ đồ tổng quan chức năng của hệ thống

***1.4.1 Chức năng chính***

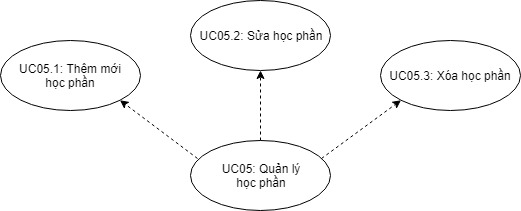
****

*Hình 1: Sơ đồ Use-case của phần mềm xếp thời khóa biểu*

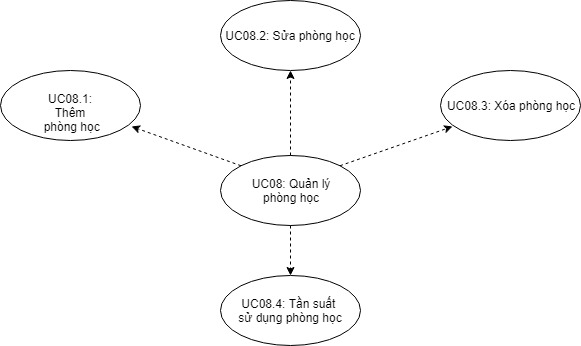
***1.4.2 Chức năng mở rộng***

**

*Hình 2: Quản lý thông tin người dùng*

**

*Hình 3: Quản lý học phần*

**

*Hình 4: Quản lý phòng học*

## Mô tả chức năng của hệ thống

***1.5.1 Các tác nhân tham gia hệ thống***

* **Người quản trị:** là người chịu trách nhiệm cho việc quản lý thông tin người dùng và phân quyền cho người dùng.
* **Phòng đào tạo:** là người phụ trách chung về chương trình học, các học phần trong chương trình học theo từng ngành, từng nhóm học. Phòng đào tạo là người duyệt thời khóa biểu cuối cùng trước khi đưa lên thời khóa biểu chính thức của toàn trường.
* **Người mở TKB bộ môn:** trong từng bộ môn khác nhau thì chỉ có người phụ trách về thời khóa biểu mới có thể mở lớp.
* **Người sử dụng:** là các giáo viên đang giảng dạy trong nhà trường. Mỗi giáo viên đều được cấp một tài khoản đăng nhập để xem thời khóa biểu của bộ môn mình cũng như thời khóa biểu toàn trường.
* **Máy in:** máy in được kết nối với hệ thống để hỗ trợ việc in ấn.

***1.5.2 Các chức năng hệ thống***

* **UC #01 – Đăng nhập:** Hệ thống cho phép các tác nhân (Người quản trị, người sử dụng, phòng đào tạo, người mở lớp) đăng nhập vào hệ thống sau khi điền thông tin tên đăng nhập và mật khẩu. Sau khi đăng nhập vào hệ thống, người dùng có thể sử dụng các chức năng của hệ thống tùy vào quyền hạn tài khoản.
* **UC #02 – Quản lý thông tin người dùng:** Chức năng cho phép người quản trị có thể quản lý được thông tin của người dùng. Chức năng bao gồm việc thêm, sửa, xóa thông tin người dùng.
* **UC #03 – Phân quyền người dùng:** Chức năng cho phép người quản trị có thể cấp quyền người dùng tùy vào mục đích của người sử dụng. Mỗi quyền được cấp sẽ có quyền hạn sử dụng hệ thống khác nhau.
* **UC #04 – Xếp thời khóa biểu:** Chức năng cho phép người mở lớp có thể tạo thời khóa biểu cho các học phần trong phạm vi phụ trách của bộ môn.
* **UC #05 – Quản lý học phần:** Chức năng cho phép phòng đào tạo có thể quản lý được các học phần trong trường học. Thông tin về học phần bao gồm: Mã học phần, tên học phần, số tín chỉ của học phần, bộ môn phụ trách học phần, … Chức năng bao gồm việc thêm, sửa, xóa học phần.
* **UC #06 – Quản lý kỳ học:** Phòng đào tạo có thể quản lý được các kỳ học của trường. Tùy vào mỗi trường mà kỳ học được chia ra khác nhau.
* **UC #07 – Quản lý nhóm học:** Chức năng cho phép phòng đào tạo có thể quản lý các nhóm học tùy vào sự sắp xếp của mỗi trường. Việc chia ra thành các nhóm học sẽ giúp cho việc quản lý dễ dàng hơn.
* **UC #08 – Quản lý phòng học:** Việc mỗi phòng học được sắp xếp như nào trong mỗi thời khóa biểu sao cho hợp lý là rất quan trọng. Chức năng cho phép phòng đào tạo có thể thêm, sửa, xóa phòng học cũng như có thể kiểm tra tần suất sử dụng phòng học theo từng mốc thời gian.
* **UC #09 – Quản lý giáo viên:** Chức năng cho phép phòng đào tạo có thể quản lý được các giáo viên trong trường. Sắp xếp giáo viên này thuộc bộ môn nào cũng như xem được những môn mà giáo viên đó phụ trách dạy học.
* **UC #10 – Quản lý thời khóa biểu:** Phòng đào tạo sau khi nhận được thời khóa biểu của từng bộ môn sẽ duyệt lại tất cả để tạo thành thời khóa biểu chung cho toàn trường.
* **UC #11 – Xem thời khóa biểu:** Người dùng có thể xem lại được thời khóa biểu sau khi đã được tạo trên hệ thống.
* **UC #12 – Tra cứu học phần:** Chức năng cho phép người dùng có thể tìm kiếm học phần mà mình muốn một cách nhanh chóng.
* **UC #13 – In ấn:** Chức năng cho phép in thời khóa biểu. Có thể in thời khóa biểu theo toàn trường, hoặc in thời khóa biểu theo từng bộ môn.
* **UC #14 – Đăng xuất:** Chức năng cho phép người dùng có thể đăng xuất ra khỏi hệ thống sau khi kết thúc phiên làm việc của mình.

## Ánh xạ yêu cầu nghiệp vụ và chức năng hệ thống

|  |  |  |
| --- | --- | --- |
| **BR** | **Mô tả** | **UC** |
| **BR1: Đăng nhập** | | |
| BR1 | Đăng nhập | UC #01 |
| **BR2: Người dùng** | | |
| BR2.1 | Quản lý thông tin người dùng | UC #02 |
| BR2.2 | Phân quyền người dùng | UC #03 |
| **BR3: Thời khóa biểu** | | |
| BR3.1 | Xếp thời khóa biểu | UC #04 |
| BR3.2 | Quản lý học phần | UC #05 |
| BR3.3 | Quản lý kỳ học | UC #06 |
| BR3.4 | Quản lý nhóm học | UC #07 |
| BR3.5 | Quản lý phòng học | UC #08 |
| BR3.6 | Quản lý giáo viên | UC #09 |
| BR3.7 | Quản lý thời khóa biểu | UC #10 |
| BR3.8 | Xem thời khóa biểu | UC #11 |
| BR3.9 | Tra cứu học phần thời khóa biểu | UC #12 |
| **BR4: In ấn** | | |
| BR4 | In ấn | UC #13 |
| **BR5: Đăng xuất** | | |
| BR5 | Đăng xuất | UC #14 |

## Lịch trình thực hiện

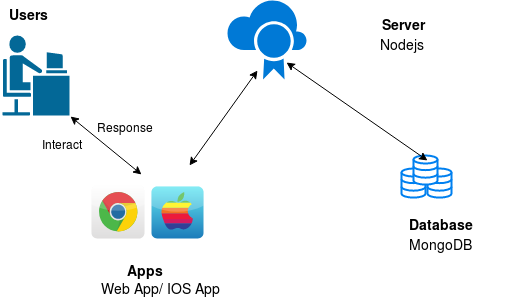
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Công việc** | **Bắt đầu** | **Kết thúc** | **Người thực hiện** | **Ghi chú** |
| **1** | **Chuẩn bị dự án** | **30/08/2019** | **06/09/2019** | **Đào Tuấn Trung**  **Tạ Minh Nhật**  **Nguyễn Hồng Quân**  **Nguyễn Tiến Dũng**  **Nguyễn Quốc Việt** |  |
| 1.1 | Lên kế hoạch | 30/08/2019 | 02/09/2019 | Đào Tuấn Trung  Tạ Minh Nhật | Kế hoạch dự án |
| 1.2 | Tạo mẫu tài liệu | 03/09/2019 | 05/09/2019 | Tạ Minh Nhật  Nguyễn Tiến Dũng | URD, tài liệu phân tích, thiết kế, báo cáo |
| 1.3 | Thống nhất công cụ dự án | 06/09/2019 | 06/09/2019 | Đào Tuấn Trung  Nguyễn Hồng Quân  Nguyễn Quốc Việt | Danh sách công cụ dự án |
|  | *Review hoàn thành giai đoạn chuẩn bị dự án* | *06/09/2019* | *06/09/2019* | Đào Tuấn Trung  Tạ Minh Nhật |  |
| **2** | **Phân tích; Thiết kế hệ thống** | **06/09/2019** | **…** | **Đào Tuấn Trung**  **Tạ Minh Nhật**  **Nguyễn Hồng Quân**  **Nguyễn Tiến Dũng**  **Nguyễn Quốc Việt** |  |
| 2.1 | Phân tích, thiết kế chức năng quản lý phòng học | 06/09/2019 | 13/09/2019 | Tạ Minh Nhật  Nguyễn Hồng Quân | Các tài liệu: đặc tả, mô tả CSDL, thiết kế chức năng, thiết kế giao diện quản lý phòng học |
| 2.2 | Phân tích, thiết kế chức năng quản lý học phần | 06/09/2019 | 13/09/2019 | Đào Tuấn Trung  Nguyễn Tiến Dũng  Nguyễn Quốc Việt | Các tài liệu: đặc tả, mô tả CSDL, thiết kế chức năng, thiết kế giao diện quản lý  học phần |

# CHƯƠNG 2: KIẾN TRÚC TỔNG THỂ, CÁC CÔNG NGHỆ SỬ DỤNG



## Kiến trúc tổng thể

***2.1.1 Sơ đồ kiến trúc***

******

***2.1.2 Mô tả kiến trúc***

Cũng giống như đa số các phần mêm hiện nay, hệ thống này được chia làm 2 phần: Client và Server. Mỗi phần đều có những chức năng và đặc điểm riêng của chúng.

* **Client:** (Web App và IOS App)
  + Đây là nơi duy nhất người dung có thể nhìn thấy, nhằm mục đích tương tác với ứng dụng này.
  + Sau khi nhận các yêu cầu từ người dùng, Client sẽ giao tiếp với Server thông qua các endpoints.
* **Server:** (Server được tạo qua Node.js và kết nối với CSDL tạo bởi MongoDB)
  + Server là nơi trực tiếp xử lý mọi request của client và trả lại những thông tin nếu cần thiết
  + Các endpoints (APIs) của server sẽ được thiết kế theo phong cách RESTful, nhằm mục đích dễ phát triển và sử dụng.

## Công nghệ sử dụng

***2.2.1 Javascript***



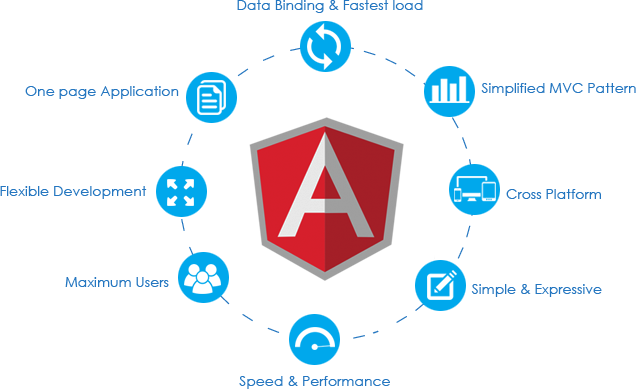
JavaScript là một ngôn ngữ lập trình đa nền tảng, ngôn ngữ lập trình kịch bản, hướng đối tượng. JavaScript là một ngôn ngữ nhỏ và nhẹ. Khi nằm bên trong một môi JavaScript có thể kết nối tới các object của môi trường đó và cung cấp các cách quản lý chúng.

Các lí do để sử dụng JS:

* Không cần một compiler vì web browser có thể biên dịch nó bằng HTML
* Nó dễ học hơn các ngôn ngữ lập trình khá;
* Lỗi dễ phát hiện hơn và vì vậy dễ sửa hơn
* Nó có thể được gắn trên một số element của trang web hoặc event của trang web như là thông qua click chuột hoặc di chuột tới
* JS hoạt động trên nhiều trình duyệt, nền tảng
* Có thể sử dụng JavaScript để kiểm tra input và giảm thiểu việc kiểm tra thủ công khi truy xuất qua database
* Hỗ trợ sẵn JSON, tiện lợi khi sử dụng noSQL (MongoDB)
* Giúp website tương tác tốt hơn với khách truy cập
* Nhanh hơn và nhẹ hơn các ngôn ngữ lập trình khác

Ngoài ra, Javascript cũng là nền tảng để tạo nên Angular và NodeJS, là hai ngôn ngữ sẽ được sử dụng chính cho phần mềm này.

***2.2.1.1 Angular***

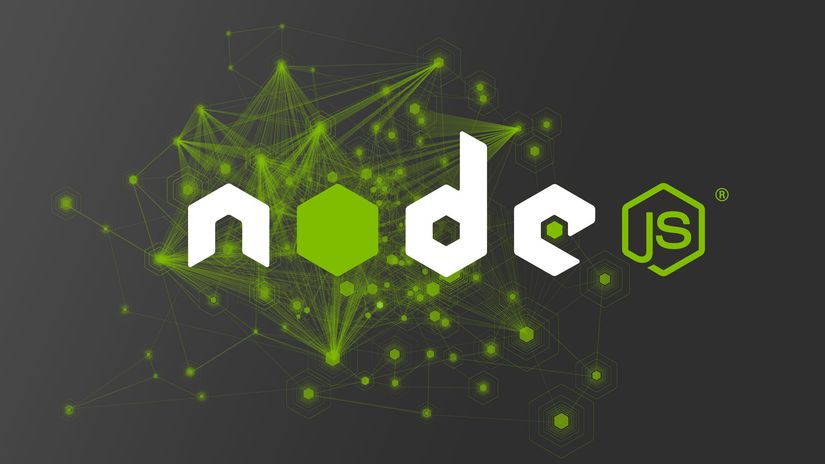


Angular là một Javascript Framework được tạo ra để xay dựng các ứng dụng web động (Dynamic Web App), nó thường được xây dựng để tạo ra các **Single Page Application** (SPA). Angular hoạt động dựa trên việc mở rộng thêm các thuộc tính (attribute) cho các thẻ của HTML. Angular là một nền tảng miễn phí và được hàng ngàn lập trình viên ưa chuộng.

Thực tế Angular có 2 phiên bản hoàn toàn khác nhau, AngularJS và Angular.

* **AngularJS:** là phiên bản đầu tiên của Angular dựa trên Javascript, được xây dựng theo mô hình **MVC** (Model-View-Controller).
  + **Model:** là thành phần trung tâm thể hiện hành vi của ứng dụng và quản lí dữ liệu.
  + **View:** được tạo ra dựa trên thông tin của Model.
  + **Controller:** đóng vai trò trung gian giữa Model và View để xử lý logic.
* **Angular:** là các phiên bản cải tiến hơn của AngularJS với lối tư duy hoàn toàn khác AngularJS. Nó được xây dựng dựa trên TypeScript.
  + **Component:** là ý tưởng chủ đạo của Angular 4. Component được Angular 4 áp dụng để tạo cấu trúc code rõ ràng và có khả năng tái sử dụng lại các component có sẵn.
  + **TypeScript:** được Microsoft phát triển để viết mã Javascript mang phong cách hướng đối tượng (Object-oriented). TypeScript mang một số tính năng như Javascript như: kế thừa, kiểu dữ liệu, hàm khởi tạo... Có thể nói, TypeScript làm cho Javascript mang phong cách rất “Java”. Về mặt bản chất nhiều cú pháp của không được trình duyệt hiểu, vì vậy nó không thể chạy trực tiếp trên trình duyệt, vì vậy cần 1 bước đó là biên dịch (compile) mã TypeScript thành Javascript thông thường. Vai trò của TypeScript như là một ngôn ngữ trung gian, nó thường được biết tới như là **Language specification** (Đặc tả ngôn ngữ).

***2.2.1.2 NodeJS***



Node.js là một nền tảng chạy trên môi trường V8 JavaScript runtime - một trình thông dịch JavaScript cực nhanh chạy trên trình duyệt Chrome. Bình thường thì ta cũng có thể tải bộ V8 và nhúng nó vào bất cứ thứ gì; Node.js làm điều đó đối với các web server. JavaScript suy cho cùng cũng chỉ là một ngôn ngữ - vậy thì không có lý do gì để nói nó không thể sử dụng trên môi trường server tốt như là trong trình duyệt của người dùng được.

Lý do sử dụng:

* Có thể tạo server trong vài dòng lệnh
* Nodes chạy đa nền tảng phía Server, sử dụng kiến trúc hướng sự kiện Event-driven, cơ chế non-blocking I/O làm cho nó nhẹ và hiệu quả.
* Có rất nhiều các packets được hộ trợ và phát triển bởi nhiều người dùng

***2.2.2 MongoDB***

***a. NoSQL***

NoSQL là 1 dạng CSDL mã nguồn mở không sử dụng T-SQL để truy vấn thông tin. NoSQL viết tắt bởi: None-Relational SQL, hay có nơi thường gọi là Not-Only SQL.

NoSQL được phát triển trên Javascript Framework với kiểu dữ liệu là JSON và dạng dữ liệu theo kiểu key và value (1 đặc trưng về dữ liệu trong JSON). NoSQL ra đời như là 1 mảnh vá cho những khuyết điểm và thiếu xót cũng như hạn chế của mô hình dữ liệu quan hệ RDBMS về tốc độ, tính năng, khả năng mở rộng, memory cache,...

Với NoSQL bạn có thể mở rộng dữ liệu mà không lo tới những việc như tạo khóa ngoại, khóa chính, kiểm tra ràng buộc, .v.v...Vì NoSQL không hạn chế việc mở rộng dữ liệu nên tồn tại nhiều nhược điểm như: sự phục thuộc của từng bản ghi, tính nhất quán, toàn vẹn dữ liệu,....nhưng chúng ta có thể chấp nhận những nhược điểm đó để khiến ứng dụng cải thiện hiệu suất cao hơn khi giải quyết những bài toán lớn về hệ thống thông tin, phân tán hay lưu trữ dữ liệu.

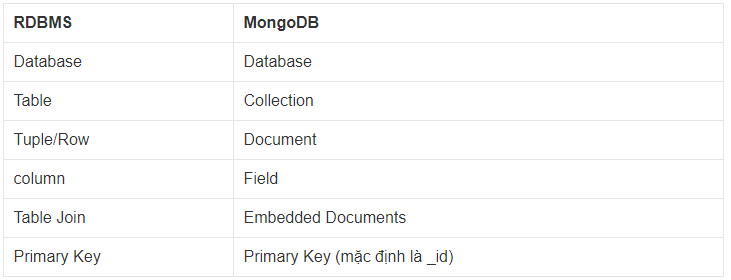
Với các CSDL quan hệ, chi phí triển khai cũng như phát triển các ứng dụng sử dụng CSDL quan hệ cũng rất tốn kém và điều quan trọng được bộc lộ rõ khi truy vấn 1 lượng bản ghi lớn trong thời gian rất dài. Tiếp đó những thiết bị cầm tay như smartphone thì không phù hợp để triển khai CSDL quan hệ vì dung lượng bộ nhớ thấp và khả năng xử lý kém hơn máy tính. Và còn rất nhiều lí do khác nữa để chúng ta quyết định việc áp dụng NoSQL vào trong dự án của mình. Bạn có thể tham khảo thêm những thông tin khác về NoSQL trên mạng.

NoSQL được sử dụng ở đâu? NoSQL được sử dụng ở rất nhiều công ty, tập đoàn lớn, ví dụ như FaceBook sử dụng Cassandra do Facebook phát triển, Google phát triển và sử dụng BigTable,....

***b. MongoDB***

MongoDB là một hệ quản trị cơ sở dữ liệu mã nguồn mở thuộc học NoSQL. Nó được thiết kế theo kiểu hướng đối tượng, các bảng trong MongoDB được cấu trúc rất linh hoạt, cho phép các dữ liệu lưu trữ trên bảng không cần tuân theo một cấu trúc nhất định nào cả (điều này rất thích hợp để làm big data).

* **Ưu điểm:**
  + Schema linh hoạt: Do MongoDB sử dụng lưu trữ dữ liệu dưới dạng Document JSON nên mỗi một collection sẽ các các kích cỡ và các document khác nhau.
  + Cấu trúc đối tượng rõ ràng: Tuy rằng cấu trúc của dữ liệu là linh hoạt nhưng đối tượng của nó được xác định rất rõ ràng. Sử dụng bộ nhớ nội tại, nên truy vấn sẽ rất nhanh.
  + MongoDB rất dễ mở rộng.
  + Không có các join: Điều này cũng góp phần tạo nên tốc độ truy vấn cực nhanh trên mongoDB, tuy nhiên có thể đạt được thông qua Aggregate.
  + MongoDB phù hợp cho các ứng dụng realtime.
* **Ánh xạ từ RDBMS (SQL) qua MongoDB**



***2.2.3 Bootstrap***



Bootstrap là một nền tảng (Framework) miễn phí, mã nguồn mở, dựa trên HTML, CSS & Javascript. Nó được tạo ra để xây dựng các giao diện Website tương thích với tất cả các thiết bị có kích thước màn hình khác nhau. Bootstrap bao gồm những cái cơ bản có sẵn như: typography, forms, buttons, tables, navigation, modals, image và nhiều thứ khác. Nó cũng có nhiều Component, Javascript hỗ trợ cho việc thiết kế Responsive của dễ dàng, thuận tiện và nhanh chóng hơn.

Bootstrap có 3 ưu điểm:

* **Dễ sử dụng:** vì Bootstrap được xây dựng trên HTML, CSS & Javascript.
* **Responsive:** Bootstrap đã xây dựng sẵn các “Responsive CSS” tương thích với các thiết bị khác nhau. Tính năng này giúp tiết kiệm thời gian cho người dùng khi tạo ra các Website thân thiện.
* **Tương thích với các trình duyệt:** nó tương thích với tất cả các trình duyệt (Chrome, Opera, Firefox ...)

***2.2.4 Angular Material Design***



Material Design là một bước phát triển của Flat Design, có thể gọi là Flat Design 2.0. Nói ngắn gọn, thay vì toàn bộ là Flat như hiện nay, Material Design là những lớp nội dung Flat được thiết kế trên những mặt phẳng xếp chồng lên nhau dưới màn hình.

Phong cách thiết kế Material Design nhắm đến những đường nét đơn giản, sử dụng nhiều mảng màu đậm nổi bật, các đối tượng đồ họa trong giao diện dường như: “trôi nổi” lên. Ngoài ra, nó còn bao gồm cả những hiệu ứng chuyển động tự nhiên khi các nút, menu hiện diện trên màn hình. Tất cả đều nhằm mang lại cho người dùng trải nghiệm mới mẻ hơn, thú vị hơn và gần giống đời thực hơn.

3 yếu tố căn bản:

* **Không gian:** không gian dưới lớp kính màn hình được mô phỏng như một không gian 3 chiều Oxyz với chiều sâu là trục Oz như trong hình minh họa. Để thể hiện yếu tố này, chúng ta sử dụng ánh sáng.
* **Ánh sáng:** là yếu tố môi trường được sử dụng nhằm thể hiện tính 3 chiều của không gian. Hệ quả của ánh sáng, hiệu ứng bóng đổ, sẽ phân định vị trí các lớp Material trong không gian theo trục Oz.
* **Material:** là những mặt phẳng có độ dày đồng nhất 1dp, Material nằm song song với mặt phẳng Oxy.

***2.2.5 Swift***

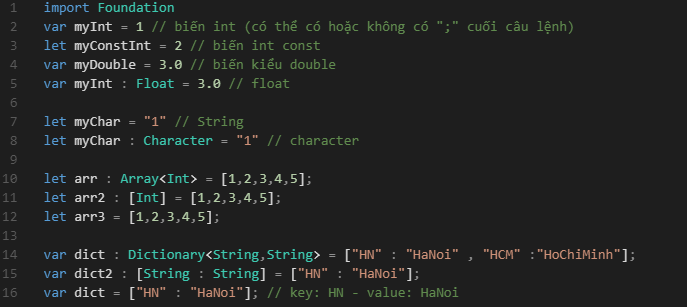
***a. Giới thiệu chung***

Swift là một ngôn ngữ lập trình hướng đối tượng dành cho việc phát triển iOS và macOS, watchOS, tvOS, Linux, và z/OS. được giới thiệu bởi Apple tại hội nghị WWDC 2014. Swift được mong đợi sẽ tồn tại song song cùng Objective-C, ngôn ngữ lập trình hiện tại dành cho các hệ điều hành của Apple. Swift được thiết kế để hoạt động với các framework Cocoa và Cocoa Touch của Apple và phần lớn mã Objective-C hiện có được viết cho các sản phẩm của Apple. Trên các nền tảng của Apple, nó sử dụng thư viện runtime Objective-C cho phép mã C, Objective-C, C++ và Swift cùng chạy trong một chương trình

Swift sẽ giúp tối ưu được phần mềm, ít xảy ra bug hơn. Ứng dụng sẽ nhẹ, tiện lợi, có tính bảo mật cao, linh hoạt trong việc nâng cấp trong tương lai. Đem lại trải nghiệm ứng dụng mượt mà hơn do được xây dựng trên nền tảng Apple khuyên dùng.

***b. Cú pháp cơ bản***

* ***Khai báo biến trong Swift:***
  + Trong Swift, ta có thể khai báo biến bằng 2 cách: sử dụng var hoặc let, let dùng để khai báo biến const, còn var là phần còn lại.



* ***Hướng đối tượng trong Swift***:
  + Hướng đối tượng nghĩa là chúng ta tổ chức phần mềm dưới dạng một sự kết hợp của nhiều loại đối tượng khác nhau mà kết hợp chặt chẽ cả về dữ liệu lẫn hành vi của chúng. Lập trình hướng đối tượng (OOP) là một phương pháp làm đơn giản hóa việc phát triển và duy trì phần mềm bằng việc cung cấp một số qui tắc. Một số khái niệm cơ bản của hướng đối tượng (OOP) là:
    - *Object*
    - *Class*
    - *Inheritance*
    - *Polymophism*
    - *Encapsulation*
    - *Abstraction*
  + Swift là một ngôn ngữ hướng đối tượng bậc cao, mang trong mình đầy đủ những tính chất cần có của một ngôn ngữ hướng đối tượng

# CHƯƠNG 3: Thiết kế API, Database



## Thiết kế kiến trúc hệ thống API

Hệ thống API của chương trình sẽ được cung cấp bởi NodeJS. Các APIs sẽ được thực hiện theo phong cách RESTful API.

Lí do đầu tiên về việc chọn cách thiết kế API là dựa vào RESTful vì chương trình này sẽ support trên đa nền tảng (hiện tại sẽ support Web App và IOS). Vì vậy, việc tách riêng việc xử lí thông tin và tương tác với cơ sở dữ liệu với việc hiển thị trong API sẽ vô cùng tiện lợi, giúp ta có thể sử dụng lại API mà không phải sửa bất kỳ chứ năng nào.

Không những vậy, bằng cách thiết kế hệ thống theo phong cách RESTful, chúng ta có thể tối ưu hóa dữ liệu bằng việc caching các dữ liệu tĩnh lại client, từ đó giảm khối lượng công việc của server, tương đương với việc có thể phục vụ cho nhiều client hơn.

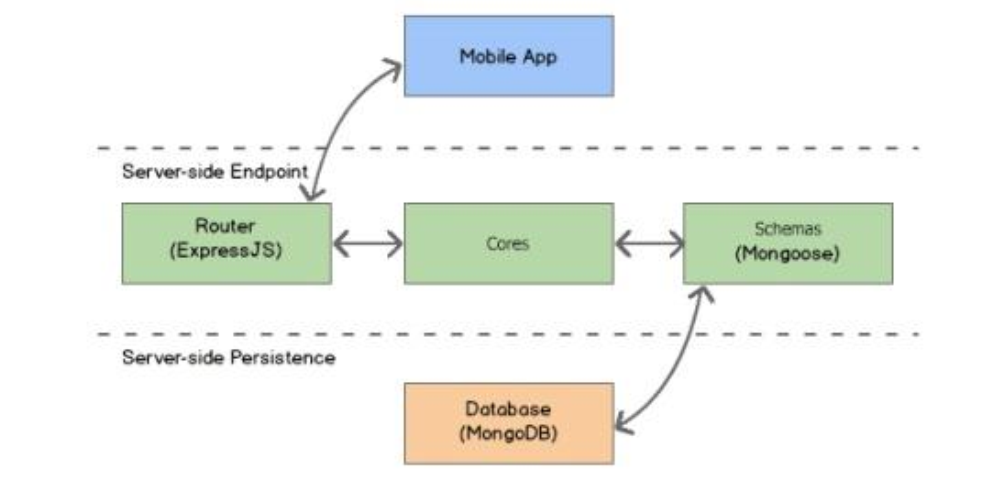
Tiếp đến, RESTful API sẽ biến các cuộc trao đổi thông tin giữa client và server trở nên phi trạng thái. Tức là, sẽ không có một chút trạng thái gì về session của client được lưu ở server. Mọi request từ client gửi lên server đều cần có đủ thông tin cần thiết để server hiểu được request đó. Điều này tương đương với việc client chịu trách nhiệm trong việc lữu trữ và xử lí các trạng thái liên quan đến thông tin. Vậy là, thêm một tính năng nữa giúp cho server được giảm tại khối lượng công việc và giảm sự phức tạp của hệ thống API đi rất nhiều.

Cuối cùng, RESTful API cung cấp cho ta một bộ các giao diện thống nhất cho API. Mỗi tài nguyên đều được tách biệt khỏi nhau thông qua một cái URL, và thông qua các method của request (GET, POST, PUT, DELETE), chúng ta có thể thao tác, xử lí tài nguyên đó. Cụ thể như sau:

Với một tài nguyên A, API của tài nguyên A sẽ bao gồm:

1. **GET** /api/A/: lấy mọi thực thể của tài nguyên A
2. **GET** /api/A/id: lấy thực thể với khóa chính là id của tài nguyên A
3. **POST** /api/A/: tạo một thực thể tại tài nguyên A
4. **PUT** /api/A/id: ghi đè/ sửa thực thể với khóa chính là id của tài nguyên A
5. **DELETE** /api/A/id: xóa thực thể với khóa chính là id của tài nguyên A

Tiếp đến, ta sẽ nói về cách để thực sự thiết kế một bộ API cho một tài nguyên A bất kỳ:



Thông qua ORM Mongoose trên nền tảng Nodejs, chúng ta có thể tạo một schema cho một tài nguyên A bất ky. Mục đích của schema là tạo khung cho tài nguyên, giúp ta hình dung và quản lý những thực thể của tài nguyên này một cách có hệ thống và hiệu qua hơn. Thông qua Schema, Mongoose sẽ tạo thành model mà sẽ được lưu vào bên trong database noSQL MongoDB, và mỗi thực thể khi lưu thành công sẽ được gọi làm một **document**. Cũng nhờ có tính năng này, trong trường hợp ta lưu một document sai cấu trúc, mongoose cũng sẽ validate và chặn không cho lưu và thông báo cho chúng ta nếu cần thiết.

Tiếp đến, tương ứng với một schema chúng ta sẽ có một thành phần Core, tại đây sẽ là nơi xử lý trực tiếp những thực thể của tài nguyên đó như: tạo, sửa xóa,... Và cuối cùng là Router, nơi đưa ra những endpoints, hay cũng chính là URL là các API dành cho tài nguyên A này.

## Danh sách API sử dụng trong hệ thống:

Hiện tại, chưng trình đã có một bộ API documentation tại đường link sau gồm có cả examples (request, response) chi tiết và đầy đủ hơn:

***3.2.1 Cấu trúc của một request***

Một request tùy vào method được sử dụng sẽ gồm có những yêu cầu sau đây:

* **GET, DELETE**: bình thường các request dạng này đều cần đi kèm theo một token được đặt trong key “Authorization” tại Header của request để cho server biết thông tin về ai là người đang gửi request này.
* **POST, PUT:** ngoài header phải có token như request ở trên (trừ một vài API đặc biệt), ở những request này cần đi kèm theo body định dạng JSON hoặc multipart (dành cho gửi file) để đính kèm thông tin giúp cho server có thể xử lí thông tin.

***3.2.2 Cấu trúc của một response***

Response sẽ bao gồm 2 thành phần chủ chốt:

* Status Code: đây là mã của response trả về, nhằm client có thể định dạng xem request vừa rồi có thành công hay không một cách nhanh chóng. Sau đây là những code sử dụng

|  |  |  |
| --- | --- | --- |
| Mã | Tên | Mô tả |
| 200 | OK | Thông báo request đã được xử lí thành công, response sử dụng được. |
| 400 | Bad request | Thông báo request được gửi đến là không hợp lí và sẽ không xử lỉ. |
| 401 | Unauthorized | Thông báo request chưa được đính kèm thông tin người gưi request cần thiết (ví dụ: chưa login) |
| 403 | Forbidden | Thông báo request này bị cấm và không được gửi. |
| 404 | Not Found | Thông bảo request của người dùng không thể tìm thấy. |
| 406 | Not Acceptable | Thông báo request của người dùng gửi thông tin không hợp lí và server sẽ không chấp nhận |
| 408 | Request timeout | Thông báo request không xử lí được vì mất quá lâu |
| 500 | Interval Server Error | Thông báo lỗi của server, không thể xử lí request |

* Response: response được trả về sẽ đi kèm hai trường chính: “message”, “data”, gồm có message là để mô tả chung về response và data là dữ liệu được trả về.

***3.2.3 Danh sách các API***

***3.2.3.1 Giáo viên***

|  |  |  |
| --- | --- | --- |
| **METHOD** | **URI** | **Mô** **tả** |
| GET | /api/teachers/ | Lấy ra thông tin của mọi giáo viên Tham số truy vấn có thể có:   * Page (int): trang bao nhiêu * Size (int): mỗi trang có bao nhiêu thực thể * Department (string): lọc giao viên thuộc khoa nào |
| GET | /api/teachers/id | Lấy ra thông tin của giáo viên có id là id |
| POST | /api/teachers/ | Tạo một thực thể giáo viên  Body của có thể tìm trong API documentation ở phía trên để có đầy đủ chi tiết |
| PUT | /api/teachers/id | Sửa/ đè thực thể giao viên có id là id  Body của có thể tìm trong API documentation ở phía trên để có đầy đủ chi tiết |
| DELETE | /api/teachers/id | Xóa thực thể giáo viên có id là id |

***3.2.3.2 Bộ môn***

|  |  |  |
| --- | --- | --- |
| **METHOD** | **URI** | **Mô** **tả** |
| GET | /api/departments/ | Lấy ra thông tin của mọi bộ môn Tham số truy vấn có thể có:   * Page (int): trang bao nhiêu * Size (int): mỗi trang có bao nhiêu thực thể |
| GET | /api/departments/id | Lấy ra thông tin của bộ môn có id là id |
| POST | /api/departments/ | Tạo một thực thể bộ môn  Body của có thể tìm trong API documentation ở phía trên để có đầy đủ chi tiết |
| PUT | /api/departments/id | Sửa/ đè thực thể bộ môn có id là id  Body của có thể tìm trong API documentation ở phía trên để có đầy đủ chi tiết |
| DELETE | /api/departments/id | Xóa thực thể bộ môn có id là id |

***3.2.3.3 Phòng học***

|  |  |  |
| --- | --- | --- |
| **METHOD** | **URI** | **Mô** **tả** |
| GET | /api/rooms/ | Lấy ra thông tin của mọi phòng học Tham số truy vấn có thể có:   * Page (int): trang bao nhiêu * Size (int): mỗi trang có bao nhiêu thực thể * Building(string): tìm theo tòa * Floor(int): tìm theo tầng * Capacity(Object): tìm theo số lượng học sinh có thể chứa |
| GET | /api/rooms/id | Lấy ra thông tin của phòng học có id là id |
| POST | /api/rooms/ | Tạo một thực thể phòng học  Body của có thể tìm trong API documentation ở phía trên để có đầy đủ chi tiết |
| PUT | /api/rooms/id | Sửa/ đè thực thể phòng học có id là id  Body của có thể tìm trong API documentation ở phía trên để có đầy đủ chi tiết |
| DELETE | /api/rooms/id | Xóa thực thể phòng học có id là id |

***3.2.3.4 Học phần***

|  |  |  |
| --- | --- | --- |
| **METHOD** | **URI** | **Mô** **tả** |
| GET | /api/courses/ | Lấy ra thông tin của mọi học phần Tham số truy vấn có thể có:   * Page (int): trang bao nhiêu * Size (int): mỗi trang có bao nhiêu thực thể * Department (string): tìm học phần thuộc bộ môn nào * Credit (Object): tìm học phần có số lượng tin chỉ |
| GET | /api/courses/id | Lấy ra thông tin của học phần có id là id |
| POST | /api/courses/ | Tạo một thực thể học phần  Body của có thể tìm trong API documentation ở phía trên để có đầy đủ chi tiết |
| PUT | /api/courses/id | Sửa/ đè thực thể học phần có id là id  Body của có thể tìm trong API documentation ở phía trên để có đầy đủ chi tiết |
| DELETE | /api/courses/id | Xóa thực thể học phần có id là id |

***3.2.3.5 Authorization***

|  |  |  |
| --- | --- | --- |
| **METHOD** | **URI** | **Mô** **tả** |
| POST | /api/auth/ | Đăng nhập, trả về một token để client lưu lại nhằm để đính kèm vào những request sau.  Body của có thể tìm trong API documentation ở phía trên để có đầy đủ chi tiết |

**Nguồn tham khảo:**

1. [**https://viblo.asia/p/tim-hieu-ve-material-design-phan-1-amoG81bnvz8P**](https://viblo.asia/p/tim-hieu-ve-material-design-phan-1-amoG81bnvz8P)
2. [**https://idesign.vn/graphic-design/material-design-nhung-nguyen-tac-can-ban-phan-1-31861.html**](https://idesign.vn/graphic-design/material-design-nhung-nguyen-tac-can-ban-phan-1-31861.html)
3. [**https://vietpro.net.vn/angular-01-gioi-thieu-angular.html**](https://vietpro.net.vn/angular-01-gioi-thieu-angular.html)
4. [**https://o7planning.org/vi/12077/gioi-thieu-ve-angularjs-va-angular**](https://o7planning.org/vi/12077/gioi-thieu-ve-angularjs-va-angular)
5. [**https://o7planning.org/vi/11695/gioi-thieu-ve-bootstrap**](https://o7planning.org/vi/11695/gioi-thieu-ve-bootstrap)